Sys Admin
the journal for UNIX and Linux systems administrators

Solaris™ Administration

Contents

A Case Study on Capacity Planning
Sponsored Content from TeamQuest Corporation · C2

Capacity Planning for the Masses — Using the SE Toolkit and Orca
Justin Buhler and Adrian Cockcroft · 1

Questions and Answers
Amy Rich · 9

The Solaris™ 9 Operating System
Sponsored Content from Sun Microsystems · 12

www.sysadminmag.com

Justin Buhler and
Adrian Cockcroft

Capacity Planning for the Masses — Using the SE Toolkit and Orca

Capacity planning is often overlooked by systems administrators until they need to justify additional hardware resources, and that's the wrong time to start. In this article, Adrian Cockcroft (author of the *Sun Performance and Tuning* book) and I intend to show that starting a capacity plan within your Solaris environment is not as difficult as it sounds. We will introduce some key principles for beginning your capacity planning and will show how we applied those principles and the tools presented in this article during the Winter 2002 Olympics.

Capacity Planning Principles

The fundamental principle behind any capacity plan begins with establishing an accurate baseline of your application's behavior over a period of time that can be easily interpreted. The following is a breakdown of this idea with the tools we use to achieve the fundamental principle.

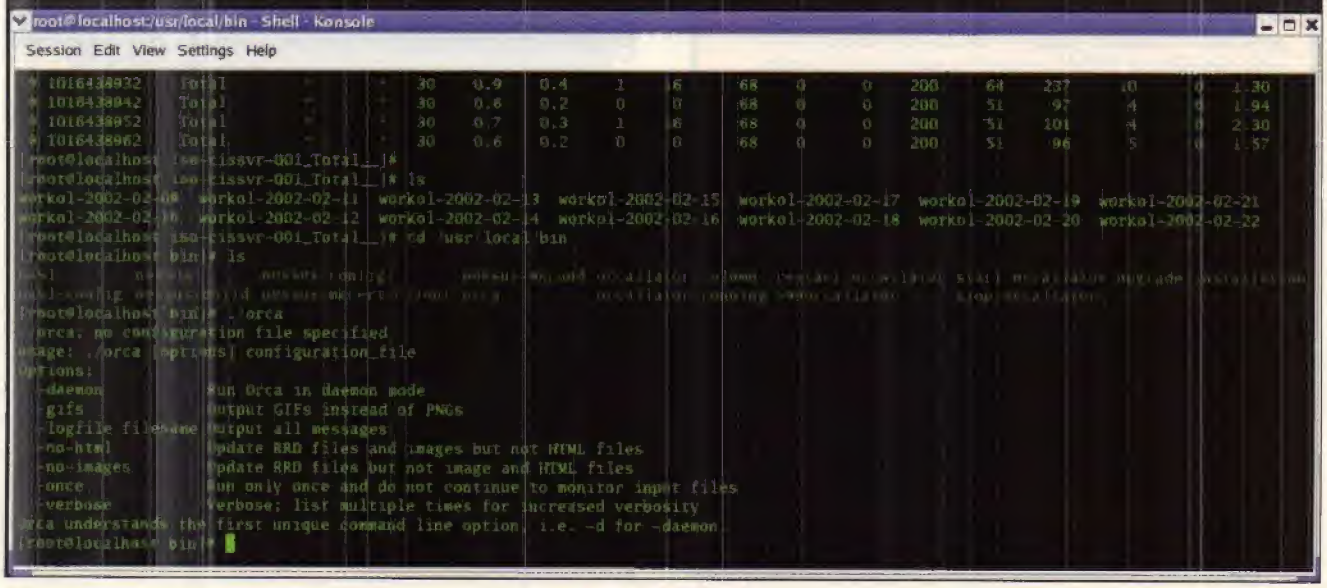
1. A capacity plan starts with accuracy, so use the right tools.

Figure 1 *Workload Example — Raw Data*

```
Time 02:30:12 nproc 7375 newproc 14 deadproc 12
wk timestamp   command    args    user count   usr%   sys% cpwt% chld%   sizeM  pf  pgwt% ulkwt%  ioK   sysc   vctx   ictx   msps
 0 1015720361    httpd       *       *     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 1 1015720361 se.sparc       *       *     2   76.4   20.5     3     0      19   0      0      0  615   6201      0    101   9.59
 2 1015720361        * tripwire      *     1    0.0    0.0     0     0      10   0      0      0    0      0      0      0   0.00
 3 1015720361        *       *      bb    10    0.0    0.0     0     0       9   0      0      0    0      0      0      0   0.00
 4 1015720361     java       *     cis     2    0.9    1.1     4     0     620   0      0      0    0    533    308      0   0.07
 5 1015720361        *       *     cis     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 6 1015720361     java       *       *     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 7 1015720361        *       *    root    18    0.0    0.0     0     0      29   0      0      0    0      0      0      0   0.00
 8 1015720361        *       *       *     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 9 1015720361    Total       *       *    33   77.3   21.6     7     0     689   0      0      0  615   6734    308    101   2.42

Time 02:30:12 nproc 7375 newproc 14 deadproc 12
wk timestamp   command    args    user count   usr%   sys% cpwt% chld%   sizeM  pf  pgwt% ulkwt%  ioK   sysc   vctx   ictx   msps
 0 1015720371    httpd       *       *     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 1 1015720371 se.sparc       *       *     2    0.7    0.2     0     0      20   0      0      0   57    104      0      0   0.00
 2 1015720371        * tripwire      *     1    0.0    0.0     0     0      10   0      0      0    0      1      4      0   0.02
 3 1015720371        *       *      bb    10    0.0    0.0     0     0       9   0      0      0    0      0      0      0   0.00
 4 1015720371     java       *     cis     2    5.4    1.5     8     0     620   0      0      0   86    847    343      6   0.20
 5 1015720371        *       *     cis     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 6 1015720371     java       *       *     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 7 1015720371        *       *    root    17    0.0    0.0     0     3      26   0      0      0    0      4     15      0   0.01
 8 1015720371        *       *       *     0    0.0    0.0     0     0       0   0      0      0    0      0      0      0   0.00
 9 1015720371    Total       *       *    32    6.1    1.7     8     3     687   0      0      0  143    956    362      6   0.21
```

Figure 2 *Orca options*



and allows long-term data views ranging from hours to years. Orca is also extended to use the SE Toolkit, and a new configuration file described below adds workollator.se workload data parsing and a set of standard plots.

Now, let's apply these principles and tools to a real-world example. We used these tools to determine the capacity for the Winter 2002 Olympics. To begin, we must clarify the location and the purpose of each of the tools. They are deployed in the following manner:

- SE Toolkit and workollator.se are installed and configured to run on the Solaris systems to be monitored.
- Orca, orca_gather.pl, and workload_extract.sh are installed on the Solaris server, where information is to be centrally collected and processed.
- The Web server is to be installed on the same server where the Orca html output is stored. It is only used to serve html, and no CGI configuration is needed.

Next, we can install the tools on the server to be monitored and start harvesting our application's stats.

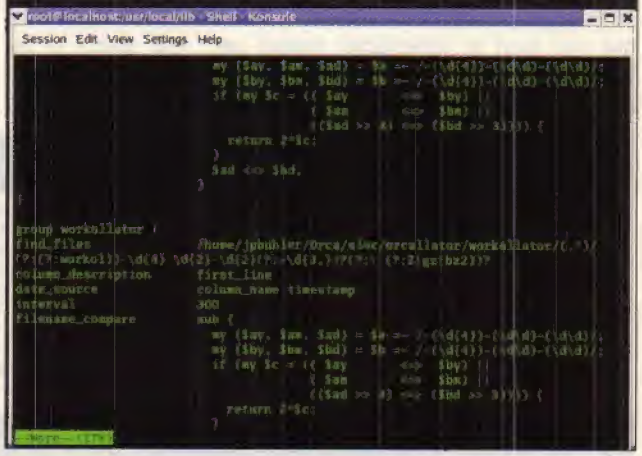
Download and install the SE Toolkit 3.3 from http://www.setoolkit.com. We will next configure a shell script to capture your system's workloads using workorallator.se on the host to be monitored. This script uses environment variables that identify the processes, which make the different workloads on the system. The environment variables are:

PW_CMD_WK# — Filter based on the command name.
PW_ARGS_WK# — Filter based on the argument strings.
PW_USER_WK# — Filter based on the user name.

Listing 1 shows an actual example from SLC Sun servers to capture the workload information from the Winter 2002 Olympic Games. Listings for this article are available from the *Sys Admin* Web site at: http://www.sysadminmag.com.

Note that the PW_CMD_3 and PW_USER_3 on lines 5 and 6 have the same workload number of _3. This is because we can stack variables and then filter. We are filtering all the Java

Figure 5 *Workollator.cfg — Workload search path example 3*



Need more SPARC?

Applications drive enterprises, not hardware. Why spend valuable resources on commodity hardware solutions, call GSH and **save up to 80%** on your Sun data center needs VOILA...

Memory

X7004A	$ 195.00
X7005A	$ 390.00
X7026A	$ 4,495.00
X7053A	$ 395.00
X7052A	$ 2,995.00
X7056A	$ 2,995.00
X7023A	$ 335.00
X7051A	$ 1,095.00
X7038A	$ 295.00
X7039A	$ 395.00

Storage

A5200 - 800G	$ 8,995.00
A5200 - 1614 G	$ 13,950.00
T3A - 327G	$ 5,995.00
T3A - 660G	$ 8,995.00
T3B - 327G	$ 9,895.00
T3B - 1G - 660G	$ 13,995.00
A1000 - 291G - 36R	$ 4,295.00
D1000 - 291G - 36R	$ 1,995.00
X9647A - T3 - 1GB	$ 4,495.00
A5200X6X9.6 TB	$ 118,500.00
3310 - 5x36 - JBOD	$ 3,995.00

Systems

E250 - 2300 - 1 G - 1x36 G	$ 2,695.00
220R - 2450 - 2 G - 1x18 G	$ 2,975.00
420R - 4450 - 4 G - 2x36 G - PS	$ 7,990.00
280R - 2900 - 2 G - 2x36 G - DVD	$ 9,995.00
E450 - 4300 - 1 G - 36 GBR	$ 2,995.00
E450 - 4400 - 4 G - 36 GBR	$ 5,995.00
V880 - WTF8-16GRF	$ 59,750.00
E4500 - 14x400/14G	$ 11,995.00
E6500 - 16x400/16G	$ 14,500.00
E3800 - 16G - RK	$ 39,950.00
E4800 - 12x750 - 24G - RK	$ 58,995.00
E4810 - 12x750 - 24G	$ 49,995.00
E6800 - 12x750 - 24G	$ 149,880.00

Invest less in SUN

Enhanced performance = More information = Improved profits = Reduced competition
Hardware won't drive the next business revolution. Information will

...invest wisely

Industry leading 36 month warranty Call and see why many FORTUNE 1000 clients use GSH as their Sun Microsystems provider.

ATL PRODUCTS · VERITAS Vplus · HITACHI DATA SYSTEMS

www.gshiis.com
email sales@gshiis.com

246 Suffolk Lane ▲ Gardner, MA 01440
(603)529-7880 ▲ Fax (603)529-7884

Sun is registered trademark of Sun Corporation
Other names may be trademarks of their respective owners

Pricing subject to change based upon availability.

gshiis intelligent integrated systems

used to store data in units. RRD will generate time-defined graphs of the data stored within the given RRD.

Orca is managed and configured from a single configuration file, called orcallator.cfg (see Figure 2). The configuration file is passed to the Orca application, which provides site-specific information, such as where to go to find the files, the name of the files, and what type of graphs to plot. The command to run after you have configured Orca is normally:

```
#>./orca -v -once  ../lib/orcallator.cfg
```

Figure 3 shows an example of the plot that is defined to generate the CPU usage of a process.

We have provided a replacement for the orcallator.cfg, called workollator.cfg, which should be passed to the Orca binary for the configuration information about how to generate the RRDs and graphs using the workload files. You must move the workollator.cfg file to the same location as the orcallator.cfg, which is usually /usr/local/lib. Next, you must modify the workollator.cfg and give it the relevant parameters specific to your site.

The following parameters tell Orca where to find or place RRDs and HTML files, as well as Orca's installation directory. Set these to the proper places. See Figure 4:

`rrd_dir` — Location in which RRD databases should be created or already exist.
`html_dir` — Location to write or update the html files.
`base_dir` — Location of the Orca installation.

The regular expression shown in Figure 5 tells Orca where to find the files and the name of those files to be loaded into RRDs. Note that there is a bug in the current distribution of the workollator script, so you need to remove the following line:

```
<a href="http://www.sun.com">
```

Processing the Workload Data

Once all of the workload data has been transferred from the Solaris servers to the Orca server, we must parse each of the daily workload logs into individual files per workload. The workload_extrctor.sh script (Listing 3) performs this task. Run this script in the directory containing all of the workload files. The result should be that additional directories are created (one for each hosts workload) under the OUTPUTDIR directory, which each contains log files for each day.

Examples of the workload directories created on the system by the workload_extractor.sh configuration are shown below. A daily log file is created under each of the host-workload directories under /opt/orca/workload/, with the name of workol-2002-02-19.

```
#> ls /opt/orca/var/workload
iso-cissvr-001___
iso-cissvr-001__bb
iso-cissvr-001__cis
iso-cissvr-001__root
iso-cissvr-001__tripwire_
iso-cissvr-001_httpd__
iso-cissvr-001_java__
iso-cissvr-001_java__cis
iso-cissvr-001_se.sparc__
iso-cissvr-001_Total__

#> ls /opt/orca/var/workload/iso-cissvr-001_java__cis
workol-2002-02-17
workol-2002-02-18
workol-2002-02-19
workol-2002-02-20
workol-2002-02-21
workol-2002-02-22
```

The format of each daily file should contain only the workload-specific data (see Figure 6).

You should now be able to run the Orca application and generate your data. The `-v` flag will enable verbose logging on all of the processing, which is useful when running Orca, especially if it is the first time to run Orca:

```
./ORCA -v -once ../lib/workollator.cfg
```

After the Orca application has completed and has returned you to the command prompt, navigate to the URL, which points to the Orca HTML files that have been generated. Once you navigate to

Sell-Rent-Lease-Buy-Upgrade
Systems, Features, Memory, Disk, Tape and more!

IBM RS/6000 - pSeries - SP2 - AS400 - SSA
Sun Microsystems, HP, Dec, SGI, Cisco Routers

From full refurbished systems to upgrades that will juice up existing hardware to its maximum potential. Let our technical sales team show you and your company how we have helped 100's of great companies. A few moments of our time will change your mind forever!

Chicago, IL - New York - New Jersey - Traverse City, MI
We will SAVE your company $$$
Elarasys.com
or call
800-787-6364

The formula used in the workload-based capacity planning model to scale the workload capacity is:

Total Sizing = Overhead + (Workload * Factor)

Overhead is everything else that is not included in the workload being modeled or Overhead = (Total - Workload). Factor is how much the workload has changed by or scale.

Figure 8 *Sample Daily Workloads (System Montitoring and Java Applicaiton)*

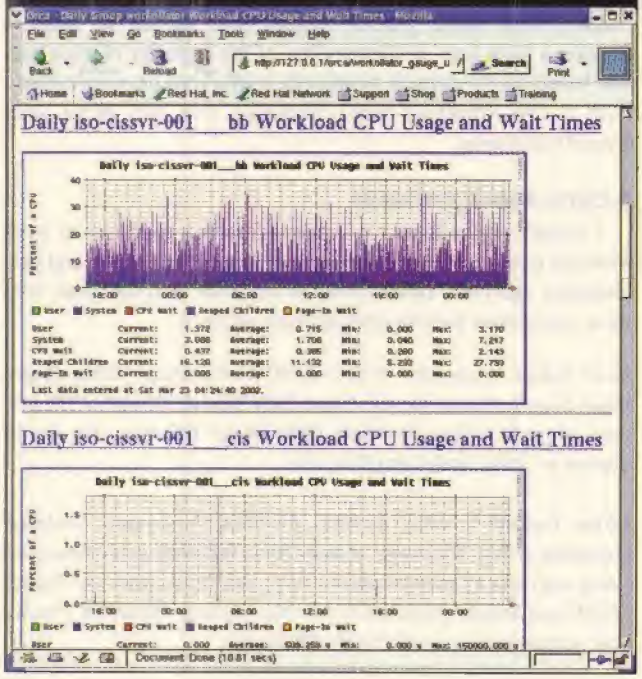
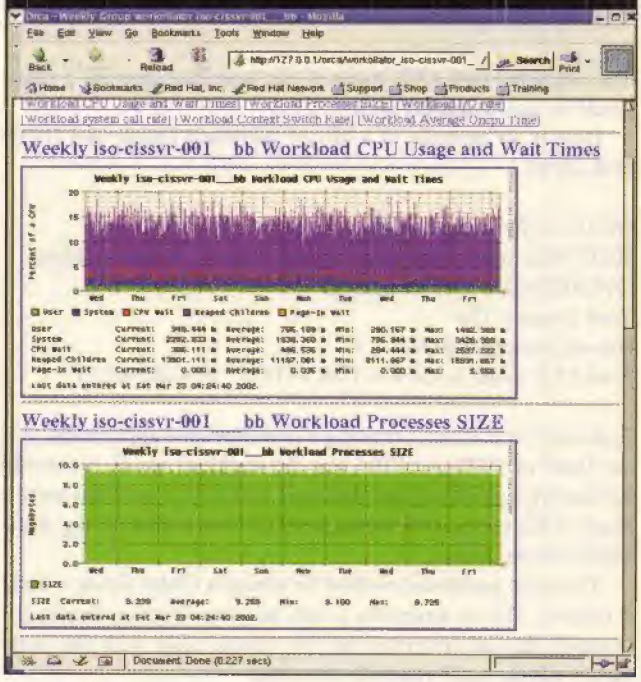


Figure 9 *Sample Weekly Workloads (System Monitoring)*



Lower your TCO

Applications drive enterprises, not hardware. Why spend valuable resources on commodity hardware solutions, call GSH and **save up to 80%** on your Sun data center needs VOILA...

...invest less in SUN

Video		
PGX 64/P		$ 265.00
C3D/S		$ 195.00
C3D/P		$ 295.00
E3DM6/P		$ 650.00
Expert 3D/P		$ 1,895.00
E3DM3/P		$ 375.00
PGX 32/P		$ 195.00
X7121A 21"		$ 395.00
X7136A 21"		$ 495.00
X7119A 19"		$ 235.00
X7124A 24"		$ 895.00
X7146A 21"		$ 695.00
Ex500 Boards		
X2602A - UG1 - 2x400/8 - 2G		$ 1,695.00
X2602A - UG3 - 2x464/8 - 2G		$ 4,695.00
Ex800 Boards		
CPUBD - 2029 - 2x900 - 2G		$ 8,995.00
CPUBD - 4047 - 4x750 - 4G		$ 3,995.00
CPUBD - 4049 - 4x900 - 4G		$ 18,900.00
CPUBD - 4169 - 4x900 - 16G		$ 22,900.00
Disk		
E250/U60/U80	18G 7.2K	$ 195.00
	36G 10K	$ 335.00
E450	18G 10K	$ 285.00
	36G 10K	$ 495.00
A5000/5100	18G 10K	$ 235.00
	36G 10K	$ 395.00
A5200	18G 10K	$ 235.00
	36G 10K	$ 495.00
	72G 10K	$ 795.00
T3	36G 10K	$ 475.00
	73G 10K	$ 795.00
Workstations		
U80-4450 - 4G - 2x36GR		$ 5,995.00
U60-2450 - 1G - 3D - 72GR		$ 2,295.00
SB1K-2750 - 2G - 2X18 G - PGX64		$ 6,295.00

Enhanced performance = More information = Improved profits = Reduced competition
Hardware won't drive the next business revolution. Information will

...invest wisely

Industry leading 36 month warranty Call and see why many FORTUNE 1000 clients use GSH as their Sun Microsystems provider.

ATL PRODUCTS VERITAS Vplus HITACHI DATA SYSTEMS

www.gshiis.com
email sales@gshiis.com

246 Suffolk Lane ▲ Gardner, MA 01440
(603)529-7880 ▲ Fax (603)529-7884

Sun is registered trademark of Sun Corporation
Other names may be trademarks of their respective owners
Pricing subject to change based upon availability

gshiis intelligent integrated systems

Amy Rich

Questions and Answers

Q I'm running Solaris 9 on a machine with one Ethernet card, but a large number of virtual interfaces. I want to use snoop to view the traffic on one of these virtual interfaces, but I'm not sure how to do that. The following command doesn't seem to work:

```
snoop -d hme0:1
```

A Snoop is used to view packets on a physical interface, not logical ones. There is no way to look only at a virtual interface. If you need to snoop for traffic on this interface separately from the others, then I suggest adding another physical network card. If you're looking for a particular kind of traffic, you can always try limiting the snoop output by using regular expressions.

Q I'm trying to build a stripped-down Solaris 8 server to act as our gateway. I tried removing the following NFS packages, since this machine should never mount anything via NFS:

```
SUNWnfscr   Network File System (NFS) client support (Root)
SUNWnfscu   Network File System (NFS) client support (Usr)
SUNWnfscx   Network File System (NFS) client support (Root) (64-bit)
SUNWnfssr   Network File System (NFS) server support (Root)
SUNWnfssu   Network File System (NFS) server support (Usr)
SUNWnfssx   Network File System (NFS) server support (Root) (64-bit)
```

It turns out that SUNWvolu claims to depend on NFS, but I figured that was just for using automounter, so I removed the NFS packages anyway. When I rebooted, `vold` no longer worked, though, so obviously there's some deeper connection. Why on earth would `vold` need NFS to mount things like CDROMs?

A `vold` treats everything like an NFS filesystem mounted under /vol. If you look in /etc/mnntab, you see the following line:

```
host:vold(pid207)   /vol   nfs  ignore,dev=4540001   1054073390
```

If `vold` mounts a CDROM, then you'll see an entry in /etc/mnntab for /vol/dev/dsk/c0t6d0/<something> as well (assuming your CDROM drive is at c0t6). The only time NFS is involved is during the actual mounting procedure. After that, the data is read by the kernel as HSFS or whatever format exists on your media.

When building secure machines, I always deinstall `vold` as well:

```
SUNWvolg          Volume Management Graphical User Interface
SUNWvolr          Volume Management, (Root)
SUNWvolu          Volume Management, (Usr)
SUNWvolux         Volume Management (Usr) (64-bit)
```

If necessary, you can always explicitly mount the CDROM by hand with the `mount` command. Doing the mount by hand doesn't require NFS at all, so all of the extraneous NFS client and server packages can be removed.

Q I'm running Solaris 8, and I fear that I've unknowingly damaged my system. There have been no changes made recently that I know of, but this machine was shut down and then rebooted due to some maintenance that was being performed on our power systems at work. When the machine was turned back on, it started to boot, and then it displayed this message:

```
mount:/usr ld.so.1 not found
```

I booted the machine from the network and then mounted /usr to look around. ld.so.1 is in fact in /usr/lib, and it looks fine. The checksums match other machines that are working ok, and the permissions match as well. Is something else broken, and the error message just misleading? Have I been hacked and not realized it? I hope you can tell me that I've somehow managed to break the machine myself (and there's an easy fix like adding a package back in or something). Help!

A You have the root filesystem and the /usr filesystem on two different slices, so it wouldn't be looking at /usr/lib before /usr was mounted. Instead, the machine is looking in /etc/lib for ld.so.1. If you boot from the net or CDROM, and look in /etc/lib, you'll probably note that this file is missing. How that happened, I'm not sure (perhaps you removed it a while ago and the machine has not been rebooted since?). /etc/lib/ld.so.1 is part of SUNWcsr, as are /etc/lib/libdl.so.1 and /etc/lib/nss_files.so.1. You may want to run pkgchk to see whether there are any other issues with SUNWcsr if you think you may have somehow damaged part of the package:

```
pkgchk SUNWcsr
```

The quick and dirty fix to get the machine up and running again is to copy /usr/lib/ld.so.1 over to /etc/lib.

Q I'm running Solaris 8 on an E450. We have a generic account that should not be able to log in, run cron jobs, etc. My understanding of the /etc/shadow file is that an entry with *NP* cannot log in but can still run cron jobs. An entry with *LK* should not be able to do either. I've modified the password entry to have *LK* in the password field, but the user can still run cron jobs. Is my understanding flawed, or did I set something up wrong or forget a step?

A Your understanding is mostly correct, with one exception. An /etc/shadow entry with *NP* can actually log in if a password is not required ‡ la ssh authorized_keys, .rhosts, etc. The issue of your locked user still being able to run cron jobs

"user@local.host". Without stickyhost, the envelope would be changed to "user@mail_hub", in order to protect against mailing loops.

Since you're not using MAIL_HUB, ruleset 5 never gets called, and that's where the LUSER_RELAY rules are (see the bottom two lines here):

```
##########################################################################
#
### Ruleset 5 - special rewriting after aliases have been expanded
###
##########################################################################
#

  SLocal_localaddr
  Slocaladdr=5
  R$+                   $: $1 $| $>"Local_localaddr" $1
  R$+ $| $#ok           $@ $1                   no change
  R$+ $| $#$*           $#$2
  R$+ $| $*             $: $1

  # deal with plussed users so aliases work nicely
  R$+ + *               $#local $@ $&h $: $1
  R$+ + $*              $#local $@ + $2 $: $1 + *

  # prepend an empty "forward host" on the front
  R$+                   $: <> $1

  # send unrecognized local users to a relay host
  R< > $+               $: < $L > $(user $1 $)  look up user
  R< $* > $+ <>         $: < > $2               found; strip $L
```

If you remove the stickyhost directive and regenerate your cf file, your LUSER_RELAY directive should work.

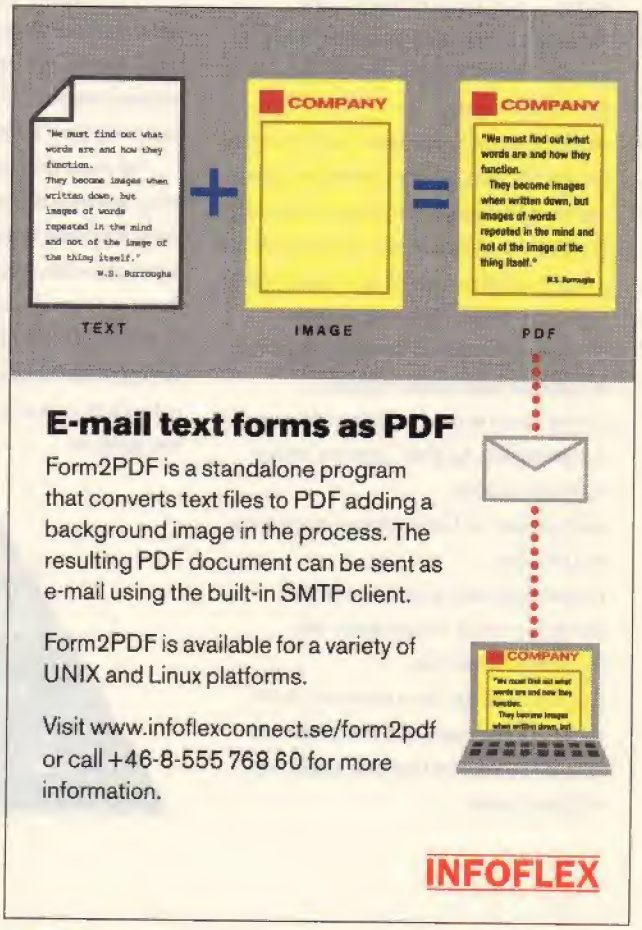
Q I'm using the gcc-3.3 package from sunfreeware.com to bootstrap my own gcc-3.3 compiler on a Solaris 9 machine. The configure goes ok, and part of the build goes ok, but then it bombs out with the following errors:

```
(cd .libs && rm -f libstdc++.la && ln -s ../libstdc++.la
libstdc++.la)
Making all in po
Making all in testsuite
mksh: Fatal error in reader: = missing from replacement macro reference
Current working directory /usr/local/src/gcc-3.3/sparc-sun-solaris2.9/ \
  sparcv9/libstdc++-v3/testsuite
*** Error code 1
make: Fatal error: Command failed for target 'all-recursive'
Current working directory /usr/local/src/gcc-3.3/sparc-sun-solaris2.9/ \
  sparcv9/libstdc++-v3
*** Error code 1
make: Fatal error: Command failed for target 'all-recursive-am'
Current working directory /usr/local/src/gcc-3.3/sparc-sun-solaris2.9/ \
  sparcv9/libstdc++-v3
*** Error code 1
make: Fatal error: Command failed for target 'multi-do'
Current working directory /usr/local/src/gcc-3.3/sparc-sun-solaris2.9/ \
  libstdc++-v3
*** Error code 1
make: Fatal error: Command failed for target 'all-multi'
Current working directory /usr/local/src/gcc-3.3/sparc-sun-solaris2.9/ \
  libstdc++-v3
*** Error code 1
make: Fatal error: Command failed for target 'all-recursive-am'
Current working directory /usr/local/src/gcc-3.3/sparc-sun-solaris2.9/ \
  libstdc++-v3
*** Error code 1
make: Fatal error: Command failed for target 'all-target-libstdc++-v3'
Current working directory /usr/local/src/gcc-3.3
*** Error code 1
make: Fatal error: Command failed for target 'bootstrap'
```

I'm not sure what mksh is, or what replacement macro reference it's talking about. Obviously, someone has gotten the compile to work because there's a package already, but I can't seem to work around this issue. Do you have any suggestions?

A You get this error when you're trying to compile some of the extra languages that come with gcc but you aren't using GNU make. You can either specify that you only want C, or you can install GNU make (or put it first in your path) and reconfigure gcc. I would suggest the latter, since various other pieces of software will also want GNU make.

Amy Rich, president of the Boston-based Oceanwave Consulting, Inc. (http://www.oceanwave.com), has been a UNIX systems administrator for more than 10 years. She received a BSCS at Worcester Polytechnic Institute, and can be reached at: qna@oceanwave.com.



The best gets better. The Solaris 9 Operating System provides a foundation for Network Identity Management through integration of the Sun™ ONE Directory Server and the Java 2 Platform, Enterprise Edition (J2EE) technology-based Sun ONE Application Server.

performance and scalability of the Solaris 9 OS, customers can immediately see an improvement in the price/performance ratio of their servers, and thus, increase the return on their investments.

The Solaris 9 Operating System offers the capabilities to scale and support massive systems and applications:

- One million simultaneous processes on a single system
- Up to 128 CPUs in a single system and 848 CPUs in a clustered environment
- Support for up to 576 GB of memory
- More than four billion network connections
- Up to 252 TB file systems with Sun StorEdge™ QFS
- IPv6, enabling a 128-bit IP address space
- A 64-bit Java virtual machine

In addition, Solaris 9 software introduces a group of new features designed to increase system performance and scalability.

Solaris 9 Threads Model

By automatically utilizing an enhanced multi-threading library, applications on the Solaris 9 platform demonstrate greater scalability and an overall performance increase of up to 4 times. There are no changes to the interfaces and no need to recode applications.

Solaris 9 Memory Optimizer

- **Multiple Page Size Support.** Applications can use multiple page sizes ranging from 8 KB to 4 MB for different memory segments. This increases performance by enhancing resource efficiency and reducing overhead. The binary is left untouched, and page sizes are dynamically changed as an application executes.
- **Advanced Page Coloring.** The Solaris 9 Operating System includes enhancements to the algorithm that control virtual/physical pages and how they are reached. As a result, system performance is increased for particularly heavy user loads.
- **Memory Placement Optimization.** Solaris 9 software is designed to optimize memory management in the way that best suits the particular servers on which it is running.
- **Memory Allocation.** Solaris now allocates memory to user-level applications much faster, and enables users to more easily debug memory leaks in their applications.

UNIX File System (UFS) Enhancements

- **UFS Performance Enhancements.** For databases created on a UFS file system, UFS Concurrent Direct I/O provides near raw device performance, leading to an 87-percent improvement in TPC-C measurements on OLTP workloads. UFS with logging enabled ensures fast file system recovery and outperforms non-logging UFS. Administrators can now get the benefits of logging with UFS without having to worry about the effect on performance. The UFS file system is also compatible with the standard EFI disk label.
- **Multiterabyte UFS.** For 64-bit SPARC platforms, UFS now supports much larger file systems, up to 16 TB in size.
- **mkfs.** Enhancements to the mkfs command have dramatically reduced the time it takes to create a UNIX file system; you can now create a UNIX file system 96 times faster.
- **64-Bit Java Virtual Machine.** With Java 2 Platform, Standard Edition (J2SE™) 1.4 software, users benefit from full 64-bit support in the Java HotSpot™ virtual machine (VM). This feature, combined with the Java HotSpot Server VM code optimizer, has significantly improved Java Servlet performance.

The first implementation of Solaris Containers is Solaris 9 Resource Manager software, which is integrated into the Solaris 9 Operating System. It consists of a set of resource management and network quality-of-service features. Solaris 9 Resource Manager enables administrators to allocate system and network resources to multiple users, groups, or applications to provide more predictable service levels. Customers can set and enforce policies that control and monitor how resources are used, and generate extended accounting information for billing or chargeback purposes. Solaris 9 Resource Manager software redefines the traditional single application system model and offers a better solution by enabling server consolidation to reduce service costs while delivering more predictable service levels.

Data Management

In the Solaris 9 platform, new features are introduced to help manage the data that gives a company its competitive advantage.

- **Solaris Volume Manager.** The Solaris Volume Manager storage management tool, which now includes a GUI, is integrated into the Solaris 9 Operating System. It enables users to manage large numbers of disks into logical volumes. The Solaris Volume Manager has been enhanced to support disks larger than 1 TB in size. Solaris Live Upgrade makes it much easier for customers to adopt Solaris Volume Manager because Solaris Live Upgrade greatly simplifies migration from other volume managers to Solaris Volume Manager. Reconfiguration Coordination Manager (RCM) support is integrated in Solaris Volume Manager, allowing the safe removal of disks through Dynamic Reconfiguration (DR) requests.
- **Soft Disk Partitions.** The soft disk partitions feature increases the number of file systems per device from eight to thousands. Partitions can also be created on top of previously defined logical volumes, giving administrators additional flexibility in configuring and managing the volumes.
- **UFS Snapshot.** With UFS snapshot, the Solaris 9 Operating System provides an online backup mechanism by creating a point-in-time image of the file systems. It helps eliminate downtime or offline time previously required to guarantee a consistent backup.

System Management

- **Administration.** The Solaris 9 Operating System provides a wide range of administration tools that assist both user and system administration tasks. It provides command-line tools and GUIs for managing users, resources, and disk storage.

  Solaris Patch Manager offers the most comprehensive patch management features for the Solaris Operating System. Administrators now can analyze the patch state of a system and automatically download the recommended patches. They are provided with the install order necessary to accommodate patch dependencies, and can use the tools on local and remote systems. All patches delivered via Solaris Patch Manager are digitally signed, helping ensure that the patches are from Sun and have not been altered in transmission.
- **Monitoring and Management.** Sun Management Center provides a powerful, easy-to-use single management point for all Sun servers and storage, independent of geographic location. System administrators can perform remote system configuration, performance monitoring, and isolate hardware and software faults through a single interface. Sun Management Center easily integrates with enterprise management frameworks, and provides a central facility for managing events and alarms, automated responses, and diagnostics.

Solaris Provisioning Services

- **Solaris Web Start.** Solaris Web Start software simplifies the installation, setup, and administration of applications written for both Solaris and Java technology-based environments with point-and-click ease of use.
- **Solaris Flash and Solaris Live Upgrade.** Solaris Flash makes it easy to provision large numbers of servers having similar configurations. It enables administrators to create a single reference installation of the entire system software stack and replicate the installation on a numbers of servers. Solaris Flash reduces installation time and configuration complexity, and simplifies the process of redeployment to support different service levels as customer demands change.

  Solaris Live Upgrade is integrated to work with Solaris Flash technology. This functionality provides a mechanism to install a Solaris Flash archive on an inactive boot environment while the active boot environment is fully functional. A simple reboot will migrate the system to the updated environment when the process is finished. This significantly reduces the installation time.
- **Solaris JumpStart Software.** With Solaris JumpStart software, the Solaris Operating Sys-tem and applications placed on a central server can be used to remotely set up a Solaris system anywhere on the network.
- **Secure WAN Boot.** Sun's new secure WAN boot technology provides mechanisms that enable system administrators to boot and install new or upgrade systems over a wide area network. It further enhances system scalability by enabling administrators to remotely install multiple duplicate systems, such as Web servers or application servers, over geographically dispersed areas.
- **Sun Management Center Change Manager.** Available separately, Sun Management Center Change Manager extends Solaris platform functionality by offering advanced provisioning capabilities. It delivers a fast and easy

The Solaris ™ 9 Operating System

Linux Compatibility

In today's world of heterogeneous computing, compatibility leads to efficiency. Combining the Linux community with thousands of Solaris software developers and nearly three million Java and XML software developers, Sun provides customers with unified access to the broadest array of innovation in the industry on which to provide services. In the Solaris 9 Operating System, more Linux applications, tools, and APIs are made available.

- **API Compatibility.** Common libraries and build environments, such as libxml, glib, and GTK+, are integrated in the Solaris Operating Environment to streamline source code development across Linux and Solaris environments.
- **Application Compatibility.** Common Linux applications, such as Samba, Apache, Linux (GNU) commands, etc., are included in the Solaris 9 Operating System. The Solaris Software Companion CD has an even more comprehensive set of free software.

Now Available for x86-based Platforms

Sun extends Solaris software's value to x86 systems by providing the same benefits and features as available on SPARC® systems. Therefore, the Solaris OS x86 Platform Edition is well-suited for companies that have already invested in Solaris software expertise and commercial Solaris applications; that need a predictable release and update process with an application compatibility guarantee; and that want Solaris source code access — all while still leveraging standard x86-based system hardware and common management tools. Applications that need even higher levels of security than those found in the Solaris OS can also take advantage of Trusted Solaris software on x86-based systems.

Modern Desktop

The GNOME 2.0 Desktop takes desktop computing on the Solaris Operating System to an entirely new level. It not only provides an easy-to-use, appealing interface, but also a wealth of productivity tools and utilities to help users get their work done efficiently.

Standards

- Interface Standards: X/Open® UNIX 98
- Graphic Standards: X11, PostScript™, Display PostScript™, OpenGL®
- Desktop Standards: CDE (Common Desktop Environment), GNOME, Motif
- Object Standards: Java IDL
- Connectivity Standards: ONC™, ONC+™, NFS, WebNFS™, SMB technologies, XML
- Internet Standards: HTTP, FTP, Telnet, DNS, NTP, IMAP4, DHCP, SNMP, IPv6, IPSec, Kerberos, SASL, OCF
- Protocols: LDAP v3 IETF, RFCs 1323, 1510, 1652, 1869, 1870, 1891-1894, 1985, 1996, 2018, 2136, 2045, 2078
- Web Services Standards: Java API for XML (JAXR), ebXML, UDDI
- XML Messaging: Java API for XML Messaging (JAXM) — SOAP, ebXML, TR&P, SOAP-RP; Java API for XML-based RPC (JAX-RPC) — SOAP, WSD
- XML Processing: Java API for XML Processing (JAXP) — XSLT, SAX2, DOM2, Schema, XSLTc

System Requirements

- SPARC® 32-bit and 64-bit platforms
- x86 (32-bit) Sun and third-party platforms (see.sun.com/bigadmin/hcl for a hardware compatibility list)
- Disk Space: 600 MB for desktop systems; 1 GB for servers
- Memory: 64 MB minimum for x86 platforms; 128 MB minimum for SPARC platforms

Learn More

Get the inside story on the trends and technologies shaping the future of computing by signing up for the Sun Inner Circle program. You'll receive a monthly newsletter packed with information on the latest innovations, plus access to a wealth of resources. Register today to join the Sun Inner Circle Program at sun.com/joinic.

To receive additional information on Sun software, products, programs, and solutions, visit sun.com/software.

For More Information

To learn more about the Solaris 9 Operating System, visit sun.com/solaris.

For more information on Solaris OS x86 Platform Edition, please visit sun.com/solaris/x86

For additional technical and development information, please visit the BigAdmin™ portal at sun.com/bigadmin.

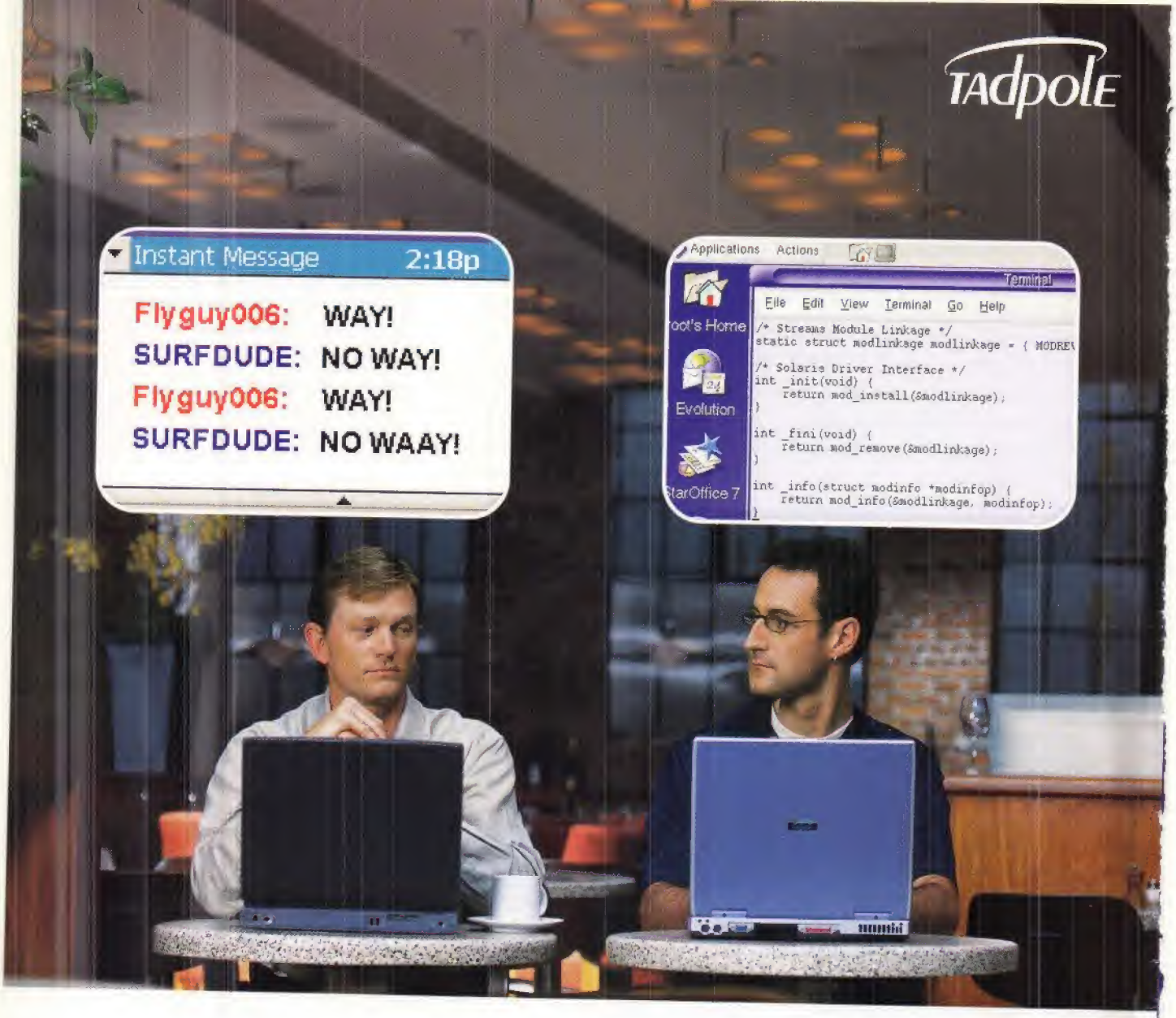
Sun Microsystems, Inc. 4150 Network Circle, CA 95054 USA **Phone** 800 786-7638 or +1 512 434-1577 **Web** sun.com

Sun microsystems
We make the net work.

Sun Worldwide Sales Offices: Australia (North, West and Coast) +61-3-9077-3800, Argentina +54-1-317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323, Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0, Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India-Bangalore +91-80-2298989/2295454, New Delhi +91-11-6106000, Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500, Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466074, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33-1, Malaysia +603-21161888, Mexico +52-5-258-6100, The Netherlands +00-31-33-451-4500, New Zealand-Auckland +64-9-976-6800, Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China-Beijing +86-10-6803-5588, Chengdu +86-28-619-9333, Guangzhou +86-20-8755-5900, Shanghai +86-21-6466-1228, Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Singapore +65-6438-1888, Slovak Republic +421-7-4342 94 85, South Africa +27-11-256-5300, Spain +34-91-596-9900, Sweden +46-8-631 10 00, Switzerland-German +41-1-908-90-00, French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-636-1888, Turkey +90-212-335-22 00, United Arab Emirates +971-4-3366333, United Kingdom +44-1276-20444, United States +1-800-555-9SUN or +1 650-960-1300, Venezuela +58-2-905-3800

SUN™ © 2003 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, BigAdmin, Java, Java HotSpot, J2EE, JSP, JVM, ONC, ONC+, Solaris, SolarisJumpStart, Solaris Enterprise Authentication Mechanism, SunOS, Sunscreen, Sun StorEdge, Trusted Solaris, and WebNFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. PostScript and Display PostScript are trade marks or registered trademarks of Adobe Systems, Incorporated, which may be registered in certain jurisdictions. OpenGL is a registered trademark of Silicon Graphics, Inc. UNIX is a registered trademark in the United States and other countries exclusively licensed through X/Open Company Ltd. X/Open is a registered trademark of X/Open Company, Ltd. Information subject to change without notice.

Printed in USA 9/03 DF16813



A serious notebook for Solaris users.

If you're thinking about buying a notebook, why settle for a Windows® laptop when you can have Solaris® to go? Because that's what a SPARC*LE*® notebook from Tadpole® gives you. With its sleek design, 2GB memory and WiFi, SPARC*LE* looks like a Windows notebook. The difference? It runs on SPARC,® meaning all the code you generate is 100% compatible across every machine Sun® makes. So now you'll be able to develop, sell, support, and use Solaris anywhere. WAY.

Enter to win a SPARC*LE* notebook. www.tadpolecomputer.com/serious

You run Solaris. So why does your notebook run Windows? Our sentiments exactly. Well, now there's SPARC*LE* from Tadpole. Starting at just $2,995.

Sun Powered Partner StarOffice ULTRASPARC DRIVEN

© 2003 Tadpole Computer, Inc. SPARC*LE* and Tadpole are registered trademarks of Tadpole Computer, Inc. All other trademarks are property of their respective owners.

way to install, upgrade, and audit the software on your systems. Change Manager enables today's business to quickly and easily provision software stacks to their servers by providing automated tools for installation or upgrade of hundreds of servers at a time — while the systems continue to operate. This can save IT costs and minimize disruptions of services. Sun Management Center Change Manager has an easy-to-use Web browser interface as well as a command-line interface (CLI) for experienced system administrators.

- **User Management.** The Sun ONE Directory Server is integrated into the Solaris 9 Operating System. It creates an LDAP-based environment that can scale to millions of users. An NIS+-to-LDAP transition kit is available to help ease migration from NIS+ to LDAP. Also, the secure LDAP client has been enhanced to support various encryption mechanisms, including DIGEST-MD5 and SSL, enabling secure password management through the Sun ONE Directory Server.

The Solaris 9 OS Is Secure

The Solaris Operating System has been built to securely interconnect with other systems and be more secure from the viruses and worms that plague software designed without security and networking in mind. These aspects of network and platform security are critical — whether building an enterprise network or providing services to millions of users over the Internet. The Solaris 9 OS includes a number of new security features.

Network Security

The network security provided by the Solaris 9 Operating System helps ensure secure authentication over the network, secure remote access, secure network connections that can't be snooped, and protection from network based attacks.

- **Solaris Secure Shell.** Solaris Secure Shell software enables strong authentication of both the client and server machines as well as users for use in remote access solutions. It also provides encryption for privacy using the 3DES, AES and Blowfish algorithms. It is compatible with other Secure Shell protocol implementations.
- **IPSec With Internet Key Exchange (IPSec/IKE).** The IPSec standard provides data integrity, privacy, and authentication of network traffic between servers for both IPv4 and IPv6 networks. IKE provides a standards-based mechanism for exchange of encryption keying material and digital certificates for use on IPv4 and IPv6 networks. Encryption is transparent to applications, and as such requires no changes to the customer experience. IPSec/IKE features DES, 3DES, AES (256-bit), and Blowfish (448-bit) encryption — all approved for export and use worldwide — and is compatible with other IPSec/IKE implementations.
- **SunScreen™ 3.2. Software.** SunScreen 3.2 software is a high-speed, stateful packet-filtering firewall that offers advanced features that protect a single system or an entire network of servers. It is now included with the Solaris 9 Operating System at no extra charge.
- **Kerberos Single Sign-On Environment.** The Solaris Enterprise Authentication Mechanism™ server and client is included in the Solaris 9 Operating System to provide single sign on capabilities for servers and applications. Kerberos-enabled versions of telnet, r* commands and more are available as a free download from the Sun Download Center.
- **TCP Wrappers.** Based on open source, TCP Wrappers provides a means of protecting your server from incoming traffic. Connections can be limited by DNS domains, IP addresses, or by substituting wild cards for part of the domains or addresses.

Platform Security

The platform security features in the Solaris 9 Operating System support server and cluster hardening through:

- Pluggable authentication modules (PAM): flexible security standards
- SSL-encrypted LDAP authentication for native login
- Role-based access control (RBAC)
- Kernel Pseudo Random Number Generator
- Nonexecutable program stacks
- Modular software packaging
- Extensive higher performance auditing with XML output
- Pluggable algorithms for password encryption (Crypt, MD5, and Blowfish included)
- Smart card authentication support
- Secure by default file permissions
- Free Solaris Fingerprint Database for file integrity verification
- Free Solaris Security Toolkit for proven system hardening

Compatibility

Compatibility is one of the hallmarks of the Solaris Operating System, and is key to enabling customers to move up the hardware product line without having to port or recompile their applications. Solaris software supports a public application binary interface (ABI) which guarantees that conforming applications will run on all Sun servers without modification.

SolCAT: Solaris Compatibility Assurance Toolkit

The Solaris Compatibility Assurance Toolkit (SolCAT) is a collection of tools and services, including the Sun Guarantee Test Suite and the Certification Test Suite, to help customers and ISVs easily and seamlessly ensure that their applications will run on the latest version of Solaris software.

The Solaris 9 OS Is Available

With businesses operating around the clock and around the globe, organizations no longer know when their customers might demand their services. The reliability of the Solaris 9 Operating System increases users' confidence that their long-running and resource-intensive applications will execute without interruption.

The Solaris 9 Operating System is designed with a small, compact kernel that limits the exposure to errors that can crash a system. It is also designed with a clear distinction between the kernel, shared libraries, and applications to further limit the impact of application failure. To deliver mainframe reliability with Internet agility, the Solaris 9 OS raises the bar for reliability, availability, and serviceability (RAS) by strengthening the following characteristics in every aspect of the development process.

Robust

- **Dynamic System Domains.** The Solaris 9 Operating System provides failure containment and high-level control over system resource allocations in electronically isolated partitions. Partition boundaries can be adjusted on the fly to rapidly adapt to changing workloads, or on a scheduled basis to enable resource shift between applications. This helps improve service levels economically.
- **Solaris Containers.** Solaris Containers isolate software applications or services using flexible, software-defined boundaries. If a fault occurs in a user-level process, the container boundary would then prevent propagation of the failure to other containers.
- **Kernel and User Mode Separation.** With the kernel occupying a protected address space, and the user-level libraries and applications occupying separate user address spaces, it is extremely difficult for a user error to cause a system failure. This significantly enhances system availability.
- **Kernel and Device Driver Hardening.** The kernel and device drivers are further hardened by identifying and eliminating panics and kernel memory leaks. The new device driver testing framework enables developers to stress drivers and simulate hardware failures to further validate that the drivers can handle unforeseeable circumstances.

Recoverable

- **File System Journaling.** By reducing file system check times during reboot, the Solaris 9 Operating System increases recoverability.
- **NFS Failover.** With NFS Failover, client systems can retry network file access on an alternate server when the primary server fails.
- **Network Multipathing.** The Network Multipathing feature allows multiple network connections to provide load spreading and failover.
- **Sun StorEdge Traffic Manager.** The Solaris 9 Operating System supports multiple paths for I/O devices, such as SCSI- and Fiber Channel-accessible storage, for load balancing and failover.

Manageable

- **Dynamic Reconfiguration.** Dynamic Reconfiguration enables system configurations to be changed without rebooting, minimizing planned and unplanned downtime.
- **Solaris Live Upgrade.** The Solaris Operating System can be upgraded while the system is still running, significantly reducing the usual service outage time associated with standard upgrades. Enhancements to the Solaris Live Upgrade Boot Environment allow Solaris JumpStart™ software to automatically create boot environments during install time, resulting in a faster Solaris Live Upgrade copy process.

Observable

- **Modular Debugger Framework.** The Modular Debugger Framework is an extensible utility that enables low-level debugging and editing of the live OS.
- **Kernel Analysis Tools.** The Solaris 9 Operating System provides a comprehensive set of "on-the-fly" kernel analysis tools. kstat, lockstat, prstat, and cpustat provide statistics and parameters for the kernel, file locks, process status, and CPU utilization respectively. truss can be used to wrap an application so that all of its system calls can be observed externally.

Availability With Sun Cluster 3.0

Sun Cluster software, available separately, provides continuous access to services via Global Network and File Services. Data, networks, and devices are available to all domains in the SunPlex systems as well as to applications running on any domain.

The Solaris 9 OS Is Manageable

As IT infrastructures grow increasingly larger and more complex, IT organizations must optimize for efficiency. Deploying and redeploying servers is a daily event. To stay ahead, businesses today must manage IT growth, while at the same time reducing complexity and management costs. The Solaris 9 Operating System provides a rich set of management facilities that can simplify the process of securely installing and deploying the software stack, resulting in lower cost of operation.

Solaris Containers and Resource Management

Solaris Containers create an execution environment within a single instance of the Solaris Operating System, and provide full resource containment, fault isolation, and security isolation. This common approach simplifies service provisioning and makes it easier to consolidate applications onto fewer servers without concern about resource constraints, fault propagation, or security.

The Solaris™ 9 Operating System

Increase service levels. Decrease costs. Reduce risks.

Sun microsystems

Key feature highlights

- Integration of the Sun™ ONE Directory Server and the J2EE™ Sun ONE Application Server in the Solaris™ 9 Operating System provides the foundation for Network Identity Management.
- Delivers the performance and stability to meet production database and file system requirements — with no incremental costs.
- Provisioning and change management provides secure installation and deployment of software stacks.
- Solaris Containers, combined with Dynamic System Domains, enable customers to get higher utilization of their system resources.
- Provides out-of-the-box security solutions and a fully integrated suite of security services to deliver the highest levels of security.
- Champions the RAS lifestyle into every step of software development process.
- Configuration services and Patch Manager, combined with SunPlex™ systems, deliver high service levels.
- Applications run faster without recompiling and recoding.
- Compatibility with previous versions and built-in Linux compatibility offer investment protection.
- The Solaris OS has been extended to x86 servers, giving customers the freedom to choose the solution that best meets their business needs.

Today, businesses are rethinking how they create, manage, extend, and ultimately deliver information technology (IT) services with greater functionality and reduced cost and complexity. Managing data center complexity from a services perspective lets businesses focus on choosing the right solution for the job at hand, rather than managing individual systems.

Since its inception in 1982, Sun's vision and strategy has been the same: connect everything through network computing. Sun continues to leverage open standards and technologies, innovate on top of them, and create the types of systems customers demand. The foundations for this revolution are Sun's Java™ Enterprise System software, Java technology, N1™ software, and the Solaris™ Operating System. Providing the most scalable product line packed with features, tighter integration, more complete testing, and the highest security levels for general purpose servers, Sun gives companies the freedom to choose the systems and software that best meet their business needs.

For more than ten years, the Solaris Operating System (OS) has delivered the power, massive scalability, high levels of security, and mainframe-class functionality that companies demand. It's the leading UNIX® environment — and the choice for powering enterprise networks that need to deliver information to networked users at any time, any place, on any platform.

Sun is now bringing the industrial strength of the Solaris OS and the military-grade security of the Trusted Solaris™ OS to x86-based systems. Providing the same functionality through the same source code, the Solaris and Solaris x86 environments deliver stability and open source innovation on both UltraSPARC® and x86-based systems.

The Solaris 9 OS Is Scalable

Companies offering Web-based services can no longer predict how many customers might visit their sites. Today's companies need the ability to grow to support millions of new customers overnight.

The Solaris 9 Operating System is the third major release of the complete 64-bit computing environment tuned for Sun's powerful line of highly scalable 64-bit servers. The multithreaded, fully preemptible kernel delivers much faster performance for core system functions and enterprise applications. And with the new increased

was actually a bug that was recently fixed, though. Ironically, because of the fix, various people have filed bug reports with Sunsolve saying that what was previously functioning has mysteriously stopped working. A couple reports state that the "fix" happened in 108993-18, so I'd try installing the latest 108993 patch and see if that helps.

Q This is sort of an odd request, but is necessitated by some testing that our development group would like to do. We're running Solaris 8 on a number of different UltraSPARC machines. We need make time appear to go faster or slower on these machines depending on the test. We don't need to muck with the hardware clock, just have the time/date change according to the rate we want. With all of the Y2K testing that had to happen several years ago, I presume someone must have come up with a product that does this. It would be great if you could point me at such a beast.

A If you're only concerned with the time and date as reported by the OS, then you need look no farther than your base Solaris install. The /usr/bin/date command takes an optional flag of -a to skew the system clock one way or another. From the man page:

> -a [-]sss.fff Slowly adjust the time by sss.fff seconds (fff represents fractions of a second). This adjustment can be positive or negative. The system's clock will be sped up or slowed down until it has drifted by the number of seconds specified. Only the superuser may adjust the time.

Q We're running Solaris 9 with a bunch of third party software installed. Unfortunately, this means we have overlapping man pages in /usr/share and /usr/local. Our standard system path has /usr/local before /usr/bin, for example, so most of the time people want the corresponding man pages in /usr/local. Occasionally, especially when writing shell scripts and explicitly using binaries in /usr/bin, people will want the man page for the stock Solaris command.

A good example is the `df` command. Because the MANPATH has /usr/local/man before /usr/share/man, people who run `man df` always get the FSF man page. Just calling nroff with the path of the system man page doesn't work because Sun no longer does their man pages in roff format. How can people access the man page for the stock Sun `df` command instead of the FSF one?

A In the case of `df`, you can actually use the section to differentiate it from the GNU `df`. For /usr/bin/df it would be:

```
man -s 1m df
```

For /usr/ucb/df, it would be:

```
man -s 1b df
```

For GNU's `df`, it would be:

```
man -s 1 df
```

There are some cases where the third party man pages and the Sun man pages will have the same section designation, though. In that case, you can specify the path you'd like to use on the command line:

```
man -M /usr/share/man df
```

Q We've finally upgraded to the latest and greatest OpenSSH (3.6.1p2) on our internal Solaris 8 machines. I can't seem to get password aging to work with the new version, and this is a big problem for us. Is there something new that needs to be configured in sshd_config or ssh_config that I've just overlooked? (I'm using the stock installed ones on the test machine.)

A Password aging on Solaris 8 is actually broken with 3.6.1. See the bug report at:

```
http://bugzilla.mindrot.org/show_bug.cgi?id=14
```

There is a third party patch you can apply to 3.6.1p2 to get it working again, though:

```
http://www.zip.com.au/~dtucker/openssh/
```

Q I've been trying to set up sendmail 8.12.9 to forward all "unknown" mail to another host. I've read through the sendmail documentation, and I think I have everything right in my mc file, but when I try to send mail to a nonexistent user, I get user unknown error instead of having the mail forwarded onto the other machine. Here's my mc file:

```
divert(0)dnl
OSTYPE(solaris2)dnl
FEATURE('nouucp','reject')dnl
FEATURE('redirect')dnl
FEATURE('use_cw_file')dnl
FEATURE('local_procmail')dnl
FEATURE('stickyhost')dnl
FEATURE('always_add_domain')dnl
FEATURE('access_db')dnl
LOCAL_USER('root')dnl
EXPOSED_USER('root')dnl
#
define('LUSER_RELAY','machine.other.domain')dnl
define('confSAFE_QUEUE', 'true')
define('confPRIVACY_FLAGS', ''authwarnings,noexpn,novrfy'')dnl
define('confTO_IDENT', '0s')dnl
define('confSMTP_LOGIN_MSG', '$j (NO UCE)')dnl
MAILER(local)dnl
MAILER(smtp)dnl
```

A It took some digging for me to find information on this particular issue, but I believe that your problem is the stickyhost feature. According to the README, stickyhost is really meant to be used in conjunction with the LOCAL_RELAY and MAIL_HUB directives. Here's the actual text:

> stickyhost This feature is sometimes used with LOCAL_RELAY, although it can be used for a different effect with MAIL_HUB.
>
> When used without MAIL_HUB, email sent to "user@local.host" are marked as "sticky" — that is, the local addresses aren't matched against UDB, don't go through ruleset 5, and are not forwarded to the LOCAL_RELAY (if defined).
>
> With MAIL_HUB, mail addressed to "user@local.host" is forwarded to the mail hub, with the envelope address still remaining

Here is an example of HTTP Server Single Workload CPU Usage:

HTTP Web Server Workload is using 15% (10 %usr + 5 %sys)
OVERHEAD is 5% total usage
Growth Factor is 5
Total CPU Usage 80% = 5% + (15% * 5)

The estimated total CPU usage if our workload grew by 5 times would be approximately 80%.

Here is an example of a Mixed Workload Oracle and HTTP Web server:

ORACLE Workload is using 45% (35 %usr + 10 %sys)
HTTP Web Server Workload is using 15% (10 %usr + 5 %sys)
OVERHEAD is 15% total usage
Total Usage = 75%
Growth Factor for BOTH is 2
Total CPU usage is 135% = 15% + (15% * 2) + (45% * 2)

Remember that if you are using a multithreaded application, you can think of 100% per CPU. For the workload above, we would be running a workload that demands 1.35 CPUs; or if this were a single CPU server, you would need to look at purchasing some additional hardware.

The more traditional method for sizing is (Total Sizing = Total * Factor). This is generally a less accurate formula because it assumes that all resources on the system scale with the same factor, which is not generally the case.

Conclusions

By using the SE Toolkit and Orca, you will quickly be able to establish a visual representation of your Sun servers' behavior and the applications that are installed. This will enable you to see and communicate a clearer and more accurate picture regarding the resource usage of all the applications. Finally, coupled with some general points, you should be able to develop a high-level understanding of the capacity of your system based on the current workloads configured on your system. With a few workload-based formulas, you can begin your capacity planning.

Acknowledgements

I extend special thanks to Adrian, who is a great guy to work with and truly an expert in the field of capacity planning and performance analysis. Thanks also to my wife and daughter who show continuous support through their patience.

Justin Buhler is currently Senior UNIX Administrator, Schlumberger, Major Event's division for the Athens 2004 Summer Olympics. His previous projects include Salt Lake 2002 Winter Olympics. He can be reached at: jpbuhler@athens2004.com.

Adrian Cockroft is Chief Architect for High Performance Technical Computing at Sun. The author of many books regarding performance and tuning and capacity planning, most notably Sun Performance and Tuning: SPARC and Solaris, *published by Sun Microsystems Press/PTR Prentice Hall. Adrian is also co-author of the* SE Toolkit.

Expand & Control Your Systems With Aurora

Aurora gives you the power to control hundreds of serial consoles from anywhere, at anytime.

XP-7R™ PCI Expansion Chassis
- Adds 7 PCI expansion slots
- PCI 2.2 compliant
- Supports 3.3V and 5V PCI cards
- PCI or cPCI (3U/6U) host interface cards available

XP-SB™ PCI Expansion Chassis
- Adds 2 (64-bit/66 MHz) or 3 (32-bit/33 MHz) PCI expansion slots
- PCI 2.2 compliant
- Supports 3.3V and 5V PCI cards
- PCI or PMC host interface cards available

ControlTower™ Console Management
- Direct console access via serial connections
- Uses a Sun Solaris host to provide network independence
- Remote access to the boot-PROM
- Maintain complete console logs
- Serial connectivity via Aurora Aries 16000P

Tel 508-588-6110 | Web auroratech.com

AURORA TECHNOLOGIES
A Carlo Gavazzi Group Company

the URL specified in your Web server configuration, you should see the index page shown in Figure 7. By clicking on the CPU usage link at the bottom of the index page, you should get the total CPU usage for each process as shown in Figure 8.

An example of a process with high chld% percentages is our system monitoring package, which executes a lot of shell scripts. See Figure 9.

Before we get into the actual planning model and examples, you'll need to be familiar with the various performance data values and how and why they are important. For a quick introduction, please see the "Performance Data Values" sidebar.

Workload Capacity Planning Model and Example

Now that we have shown how to get the workload data into a nice format, let's finish up with the formula that can be used to scale the capacity of a system.

The formula for workload-based capacity planning should be based on the workollator data, because this shows how much of the CPU is being used by the workload you are interested in rather than a system total. The system total can be considered as overhead or possible bottlenecks to the workload being scaled. Note that the workload shows what the workload uses in (%usr time) and what the system does to service the workload in (%sys time).

Introduction to Performance Data Values

Wk — This is the assigned numeric workload number ID when collecting data.

Timestamps — Timestamps are in Unix Epoch time format, which is the number of seconds after 00:00:00 UTC on January 1 1970.

Command — The workload search filter based on the command name.

Args — The workload search filter based on processes arguments.

User — The workload search filter based on the processes running user name.

Cnt — The number of concurrent instances of the running workload process.

usr% — The total percentage of time the workload spent running on the CPU executing in the developer code.*+

sys% — The total percentage of time the workload spent running on the CPU executing in the kernel servicing the system calls for your application.*+

cpwt% — The total percentage of time the workload was ready to run, but spent in the RUNQUE waiting to get back on the CPU to begin executing again.

chld% — The total percentage of CPU time of the child processes that have exited. You will see high chld% percentages if your application is forking off multiple shell scripts. This happens typically with monitoring applications. (See Figure 9 for a great example, represented as reaped children).

sizeM — The virtual address space for the application. Most long-lived processes grow their address space quickly when they first start up then stabilize. If a process's virtual memory size continues to increase, then it is quite likely that it has some kind of memory leak. This rule is useful for long-term analysis; it does not apply recently created processes.

pf — The number of page faults that had been triggered. Page faults occur when the process returns to the CPU and finds that a page in memory no longer exists and has to be reclaimed back to the memory freelist. This can indicate a memory-shortage if seen in high numbers.^

pgwt% — The total percentage of time waiting for page fault to be completed. The time spent waiting for page faults for non-executable data pages indicates I/O activity. The pgwt% can be used to indicate a memory shortage

ulkwt% — The total percentage of CPU time the user-lock from idle threads stopped on a semaphore. This is relevant on systems with databases.

Process with Blocked Threads Rule — Multithreaded processes can consume more than 100% CPU time, and can also report more than 100% of wait time. Every thread that is blocked on a semaphore will report via the microstate "user lock wait time". This can be confusing as you could see a process accumulating hundreds or thousands of wait% time in a report. The fact is that there are a bunch of idle threads; no CPU resources are being consumed. The Java runtime environment is multithreaded, and it is quite common for this to occur with Java programs.

ioK — The total number of characters in Kb read and written in read/write system calls.

sysc — The total number of system calls executed for the workload.

vctx — Voluntary context switches happen when the application leaves the CPU on its own. An example would be waiting for the system to service a disk I/O. If you see this consistently high, then it may indicate that the application is spending more time waiting for its own interrupts to be completed than processing on the CPU. Voluntary context switches occur when the process blocks in a system call to wait for something else to complete.

ictx — Involuntary context switches happen when a process is interrupted or kicked off the CPU by a high priority process or a system interrupt. High involuntary context switches can indicate that the system has lots more processes with higher priority keeping your process from getting CPU time. An involuntary context switch occurs when another higher priority process has taken over the CPU, and often occurs when the process has used up an entire time slice and had its priority reduced as a consequence. So, voluntary context switches should be expected, but involuntary context switches indicate that there is some contention for the CPU and a bottleneck exists.

msps — The milliseconds per context switch shows how long the process ran on average before it switched off the CPU. 1 seconds = 1000 milliseconds and 1 minute = 60000 milliseconds.

---

*The microstate measures of user and system CPU time don't miss anything, so if they are zero the process has definitely not run in the interval.

+You may see the CPU percentage reach over a 100% on a multiprocessor machine; the following can be read to correctly interpret (100%=1 CPU).

^Swapping happens to idle processes, so when memory is short you would expect to see one process reporting a memory shortage and different processes reporting that they have been swapped out to make space.

processes running as the cis user. Also, the 300 after the workollator.se on line 14 represents the number of seconds after which the process will wake up and capture the workload data in a daily log file, located in the directory as defined by the OUTDIR variable.

An example of the log file created on the system by the workollator.se configuration is shown in Figure 1. The file is created daily in the directory /opt/orca/workload/ with the name of iso-cissvr-001-2002-02-19.

Collecting Data Centrally Using rsync and SSH

The workload data from the servers must be collected centrally on to the Orca server for processing. In Salt Lake, we used a simple Perl script using rsync (`http://www.samba.org/rsync`) and SSH to read from a file the IP addresses and hostnames of all the servers from which data needed to be collected. The SSH account that was used to authenticate needed only read access to the workollator logs in the directory as defined in the OUTDIR variable in our cis_workollator.sh script (see Listing 2).

Installing Orca

The next step is to install Orca on a Web server machine. Orca provides an excellent "Orca installation and configuration guide", located in the contents of the package, which can be downloaded from:

`http://www.orcaware.com/orca.`

Orca needs only to know where the HTML pages are to be stored and the location of your Web server access log. This information is only used for the initial installation and can be changed in an Orca configuration file. The version we are using for our installation is orca-0.27b3.tar.gz.

Introduction to Orca and RRD

Orca is a toolkit that will graph arbitrary data into HTML documents using PNG or GIF graphics generated from an RRD (Round Robin Database). RRD is the datastore, or database, that is

Figure 6 *Java workload extraction data example*

wk	timestamp	command	args	user	count	usr%	sys%	cpwt%	chld%	sizeM	pf	pgwt%	ulkwt%	ioK	sysc	vctx	ictx	msps
4	1016382482	java	*	cis	2	4.0	1.2	6	0	623	0	0	100	0	635	363	1	0.14
4	1016382492	java	*	cis	2	3.9	1.2	8	0	623	0	0	100	0	605	362	1	0.14
4	1016382502	java	*	cis	2	4.0	1.3	13	0	623	0	0	100	0	600	357	1	0.15
4	1016382512	java	*	cis	2	4.3	1.3	8	0	623	0	0	100	0	671	365	1	0.15
4	1016382522	java	*	cis	2	4.1	1.2	7	0	623	0	0	100	0	626	362	1	0.15
4	1016382532	java	*	cis	2	4.0	1.2	12	0	623	0	0	100	0	604	358	1	0.15
4	1016382542	java	*	cis	2	4.0	1.2	7	0	623	0	0	100	0	631	362	2	0.14
4	1016382552	java	*	cis	2	3.9	1.2	8	0	623	0	0	100	0	607	362	1	0.14

Figure 7 *Sample Workload Page from Salt Lake 2002 Server*

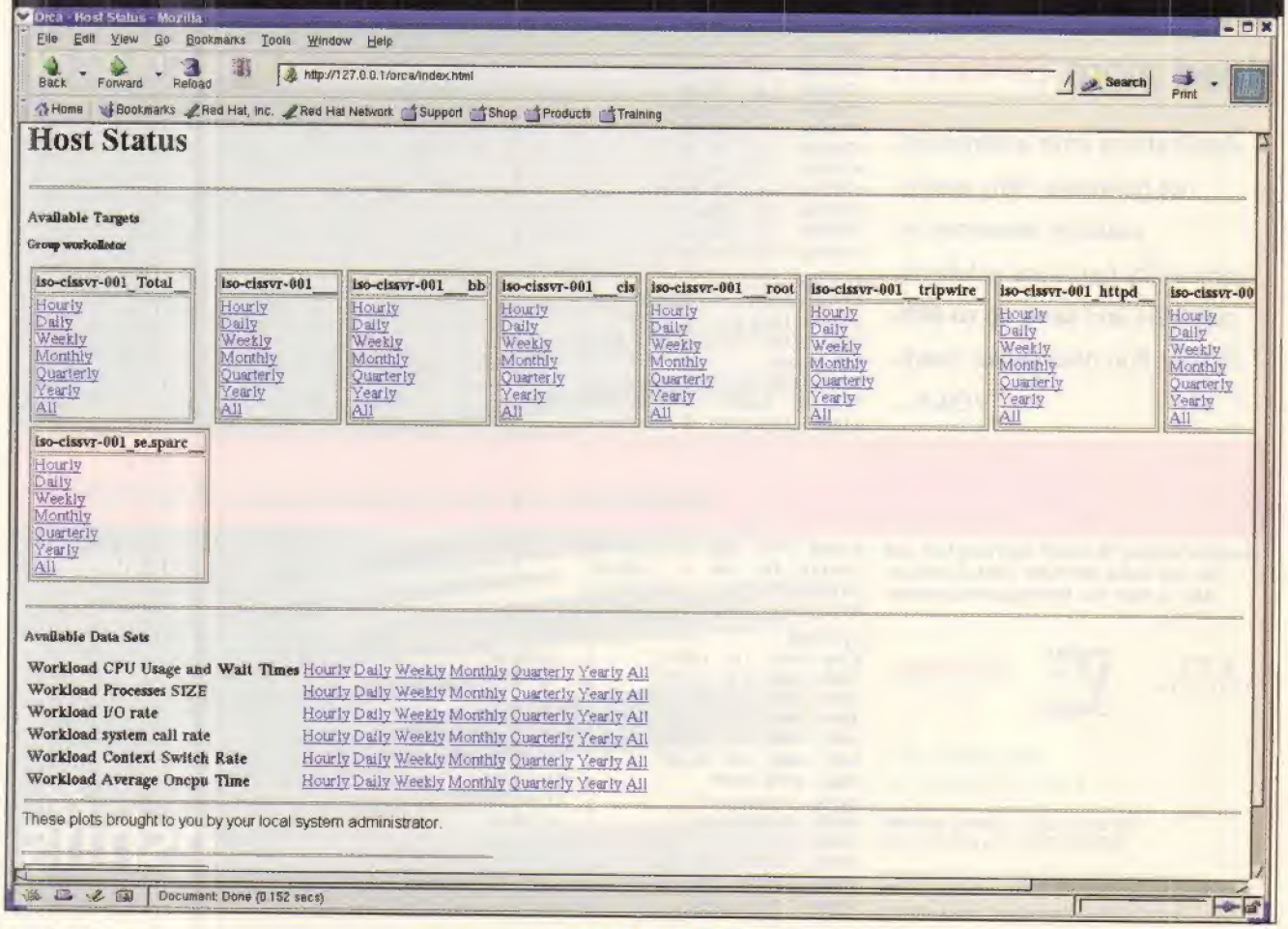


Figure 3 *Workollator.cfg — CPU usage graph syntax example 1*

```
root@dhcp-ellg01-91-12:/usr/local/lib - Shell - Konsole
Session  Edit  View  Settings  Help
plot {
title           %g Workload CPU Usage and Wait Times
source          workollator
data            usr%
data            sys%
data            cpwt%
data            chld%
data            pgwt%
line_type       area
line_type       stack
line_type       stack
line_type       stack
line_type       stack
legend          User
legend          System
legend          CPU Wait
legend          Reaped Children
legend          Page-In Wait
y_legend        Percent of a CPU
data_min        0
data_max        12800
plot_min        0
#ref             http://www.orcaware.com/orca/docs/\
                 orcallator.html#cpu_usa
%%


plot {
title           %g Workload Processes SIZE
source          workollator
data            sizeM
```

A starting point is to capture accurate system behavior information by using microstate accounting tools. This means your tools can harvest the performance data from the system without becoming one of the performance problems that needs to be investigated. The SE Toolkit developed by Rich Pettit and Adrian Cockcroft is able to accomplish this; not only are the access methods very lightweight, but the interpreter itself is written such that execution speed is very fast. SE Toolkit is available free from:

```
http://www.setoolkit.com
```

2. To make your system easy to understand, break down the system's activity.

   By breaking down the system into workload sets, you can intuitively analyze the machine in terms of categories or "slices of a pie". This categorizing or slicing is called workload analysis. Because every system runs a mixture of workloads, you just need to pick out the relevant processes and then aggregate the results. The rest is considered overhead of unplanned activity.

   The workollator.se class, which is included in the SE Toolkit 3.3, was written during the Winter 2002 Olympics, to ease the collection of accurate data about a defined workload set in a single consistent daily log. This allows you to see what each process or workload is doing during the given period of time.
3. "A picture is worth a thousand words", so make the raw data intuitive to understand.

   You need to be able to make sense of the data that you are collecting about your workloads. Looking at thousands of lines of performance data is not intuitive, even for gurus. Graphing your performance data gives you the best visual representation of all of those numbers and is much easier to decipher. One tool that enables this visualization of your performance data is Orca, which is available from:

```
http://www.orcaware.com
```

The Orca utility is written by Blair Zajac and currently is free. Orca uses the RRD for plots by Tobias Oetiker, available from:

```
http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/
```

Figure 4 *Workollator.cfg — ENV Path configuration example 2*

```
root@dhcp-ellg01-91-12:/usr/local/lib - Shell - Konsole
Session  Edit  View  Settings  Help
[root@dhcp-ellg01-91-12 lib]# more workollator.cfg
# Orca configuration file for orcallator and workollator files.

# Require at least this version of Orca.
require                 Orca 0.264

# base_dir is prepended to the paths find_files, html_dir, rrd_dir
# and state_file only if the path does not match the regular
# expression ^\\?\.{0,2}/, which matches /, ./, ../, and ~/.
base_dir                /home/jphuhler/orca/sloc/orcallator

# rrd_dir specifies the location of the generated RRD data files.
rrd_dir                 .

# state_file specifies the location of the state file that remembers
# the modification time of each source data file.
state_file              orca.state

# html_dir specifies the top of the HTML tree created by Orca.
html_dir                /var/www/html/orca

# By default create .meta tag files for all PNGs or GIFs so that the
# web browser will automatically reload them.
expire_images           1
--More--(33%)
```

A Case Study on Capacity Planning

By Mark Ehr, Senior Analyst – Enterprise Management Associates

Note: The following is excerpted from "The Value Proposition for Capacity Planning," a white paper prepared by Enterprise Management Associates. To view the paper in its entirety, visit www.teamquest.com/ema

A contractor to the US Government that provides services to many government agencies (and who requested anonymity) has a database engine that contains information on hundreds of thousands of military personnel and their dependents. Just a few years old, the system has become enormously popular, growing from two UNIX servers to nearly 200 since it began operations.

Such rapid growth was good news for the company, but bad news for its IT organization, which was scrambling to keep up with the skyrocketing demand for server capacity. To have any chance to catch up, the company needed automation and planning technology to facilitate growth. The company also needed performance reporting and tracking capabilities to ensure good service and debug performance problems.

In 2001, the company's sole server provider, Sun Microsystems, recommended TeamQuest® Model to the company's IT department. TeamQuest technicians came to the company and installed Model on six Web servers to collect data and create some sample models for IT staff. Once company officials gave the go-ahead, the full Model package was installed and operational within a few days.

> Since its installation, TeamQuest® Model has enabled the company to save millions of dollars

Since its installation, TeamQuest Model has enabled the company to save millions of dollars in over-provisioning costs, according to officials associated with the Model deployment. In one instance, Model modeling helped the company find a configuration that spread the workload across existing Sun servers, reducing the need for new servers from 30 to 20 and saving the organization millions of dollars.

Today, Model is helping the company maintain consistent levels of service across its government customer base, which is a crucial element to its business. Model enables the company to model server performance and establish service level agreements with realistic thresholds, making it easier for the organization to meet or beat customer expectations, even as its server infrastructure continues to evolve.

EMA Perspective

In a difficult economy, every IT dollar matters. The days of solving performance problems by indiscriminately buying more servers are over, and there is an excellent window of opportunity for deploying management technology that can optimize server utilization and performance. IT organizations are looking for ways to make better use of existing server capacity while reducing the need to buy more processor power.

> Through capacity planning IT organizations can realize major cost savings immediately

At the same time, it is important to note that IT budgets are tight, and that most IT organizations are not prepared to purchase new management or planning software unless they are guaranteed a clear and fast return on their investment (ROI). Today's management tools must be affordable, easy to deploy, and show strong results in a very short period of time.

Capacity planning tools meet all of these criteria. Through capacity planning, IT organizations can realize major cost savings immediately by reducing or eliminating the need to over-provision their server environments. Capacity planning and modeling technology also can guide IT through the server consolidation process, and can help set realistic goals for service levels that can be maintained over time.

Copyright © 2003 TeamQuest Corporation. All Rights Reserved. TeamQuest and the TeamQuest logo are registered trademarks in the US, EU and elsewhere. All other trademarks and service marks are the property of their respective owners.

TeamQuest®
Optimizing IT Performance